IFM 20915 - Programmation Internet II

Cascading Style Sheet (CSS)

CSS

Cascading Style Sheet (CSS)

- Définie le style d'une page web
- Beaucoup plus d'option qu'avec du HTML pure
- Facilite la réutilisation
- Facilite la travail division du travail
- Peut avoir différent style pour différent appareil

CSS

feuille de style externe

```
<link rel="stylesheet" type="text/css" href="style1.css" />
```

rel : relation entre le document actuel et le document lié

type : type de média du document lié

href : emplacement du document

media : pour quel appareil le document lié sera affiché

CSS

feuille de style interne

```
<head>
    <title> Hello World </title>
    <style>

    </style>
```

Peut avoir le CSS directement dans la page HTML

Le code doit être mis dans une balise <style> qui sera situé a l'intérieure de la balise <head>

styles en ligne

La plupart des balise on un attribut "style" dans lequel on peut mettre le CSS directement

```
<p style="">
```

CSS - Syntaxe

Sélecteur

Ce qui sera affecté par le code

Propriété

La propriété qui sera modifié

Valeur

La valeur qui sera appliqué a la propriété

```
p
{
    color:red;
    background-color:yellow;
}
```

CSS - Exemple

background-color : la couleur de l'arrière plan

color : la couleur du texte

border : la couleur de la bordure

```
<p style="background-color:red">Exemple</p>
<p style="color:red">Exemple</p>
<p style="border:red; border-style:solid">Exemple</p>
```

Exemple

Exemple

Exemple

CSS - Sélecteurs

Jusqu'à présent, nous avons seulement utiliser le sélecteur d'élément

élément	p	les éléments "p"
.classe	.bonjour	les éléments avec l'attribut class="bonjour"
#id	#titre2	l'élément avec l'attribut id="titre2"
*	*	Tous les éléments
élément, élément	th, td	les éléments "th" et les éléments "td"
élément élément	div p	les éléments "p" qui sont a l'intérieure d'un "div"
élément > élément	div > p	les éléments "p" dont le parent direct est un "div"

Notez que pour les 3 dernier, on peut utiliser des selecteurs autre que des éléments

CSS - Sélecteurs

:first-child	p:first-child	les éléments "p" qui sont le premier élément de leur parent
:first-of-type	p:first-of-type	les éléments "p" qui sont le premier élément "p" de leur parent
:last-child	p:last-child	les éléments "p" qui sont le dernier élément de leur parent
:last-of-type	p:last-of-type	les éléments "p" qui sont le derner élément "p" de leur parent
:nth-child(n)	p:nth-child(3)	les éléments "p" qui sont le 3eme élément "p" de leur parent
:nth-child(an+b)	p:nth-child(2n)	les éléments "p" qui dont l'index est un multiple de 2
:link	a:link	un lien normal
:visited	a:visited	un lien qui a déjà été visité
:hover	p:hover	un éléments "p" quand la sourie le survole
:active	p:active	un éléments "p" au moment qu'il est cliqué

CSS – Rule Cascade

Que se passe-t-il si plus d'une déclaration de style s'applique à une propriété d'un élément?

```
p {color:red;}
.para {color:blue;}
#premier {color:yellow;}
```

```
<p id="premier" class="para">Quelle couleur?</p>
```

CSS – Rule Cascade

1- Media Type

Les règles déclaré spécialement pour un type de media aura priorité

2- !important

Nous pouvons déclarer un style comme étant important avec le mots clef !important

```
p
{
    color:red  !important;
    padding : 3px;
}
```

CSS – Rule Cascade

3- Le sélecteur

3.1- l'attribut de la propriété style=""

3.2- #id

3.3- .class

3.4- élément

3.5- *

4- Attribut HTML

Par exemple la propriété "font-weight" de <b>

CSS – Rule Cascade

5- En ordre : la dernière règles écrase les anciennes

```
p { color : red;}
p { color : blue;}

<p>Je suis bleu </p>
```

Note : les règles dans des CSS externe sont traité en premier (et donc seront écrasé par les règles interne)

CSS – Héritage

Certaines propriété sont héritées par les descendant de l'élément ... et certaines ne le seront pas

Par exemple :

```
<div style="color:red;border:2px solid red">
    Par exemple
    <div>
        La couleur du texte sera héritée, mais pas la bordure
    </div>
</div>
```

Par exemple
La couleur du texte sera héritée, mais pas la bordure

Ils n'y a pas de règles pour savoir quelle propriété sera hérité, il faut soit les savoir par cœur, le tester ou regarder en ligne

CSS – Disposition (Layout)

Float : Spécifie qu'un élément doit être placé sur le côté gauche ou droit de son conteneur, ce qui permet aux texte et aux éléments inline de l'entourer

none : Valeur par défaut

left : L'élément va flotter à la gauche

right : L'élément va flotter à la droite

Lorem ipsum dolor sit amet,
tincidunt ut laoreet dolore ma
nostrud exerci tation ullamco
autem vel eum iriure dolor in

Lorem ipsum dolor sit amet, consectet
euismod tincidunt ut laoreet dolore ma
minim veniam, quis nostrud exerci tati
aliquip ex ea commodo consequat. Du
vulputate velit esse molestie consequa
at vero eros et accumsan et iusto odio
zzril delenit augue duis dolore te feugait nulla facilisi.
adipiscing elit, sed diam nonummy nibh euismod tinc

g elit, sed diam nonummy nibh
erat volutpat. Ut wisi enim ad
er suscipit lobortis nisl ut
eum iriure dolor in hendrerit in
olore eu feugiat nulla facilisis
. blandit praesent luptatum
si.Lorem ipsum dolor sit amet, consectetuer
ncidunt ut laoreet dolore magna aliquam erat

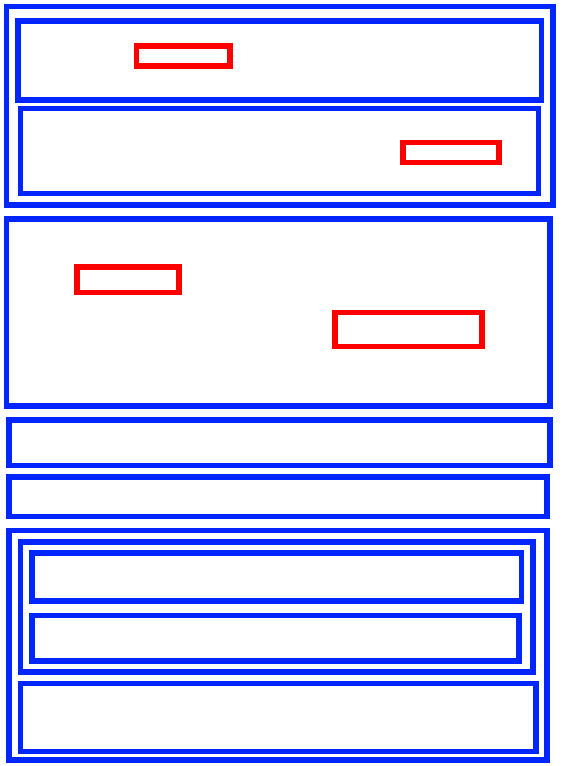
CSS – Disposition (Layout)

Ils y a deux type d'élément HTML, Block & Inline

Les éléments block commence toujours sur une nouvelle ligne et occupe tout la largeur disponible

- les block enfant sont contenue a l'intérieur de leur parent
- les block frère sont un pardessus l'autre

Les éléments inline ne commence pas sur une nouvelle ligne et occupe seulement la largeur nécessaire



CSS – Disposition (Layout)

Block

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>	
<dl>	<dt>	<fieldset>	<figcaption>	<figure>	<footer>	<form>
<h1>-<h6>	<header>	<hr>		<main>	<nav>	<noscript>
	<output>	<p>	<pre>	<section>	<table>	<tfoot>
	<video>					

Inline

<a>	<abbr>	<acronym>	<b>	<bdo>	<big>	
<button>	<cite>	<code>	<dfn>		<i>	<img>
<input>	<kbd>	<label>	<map>	<object>	<q>	<samp>
<script>	<select>	<small>			<sub>	<sup>
<textarea>	<time>	<tt>	<var>			

 & sont souvent utilisé pour le CSS

CSS – Disposition (Layout)

On peut changé le comportement des éléments avec la propriété "display"

Display

- block : Affiche un élément comme s'il s'agissait d'un élément block
- inline : Affiche un élément comme s'il s'agissait d'un élément inline
- inline-block : Affiche un élément comme s'il s'agissait d'un élément inline MAIS on peut appliquer des valeur width et height
- none : L'élément est complètement supprimé

Un exemple courant consiste à créer des éléments <li> en ligne pour les menus horizontaux

CSS – Disposition (Layout)

Jusqu'à présent, il n'était pas possible de positionné des éléments block horizontalement (sans l'aide de tableau)

Le CSS nous permets de contrôler la position et la disposition des éléments avec beaucoup plus de précisions

CSS – Disposition (Layout)

Top, Left, Bottom, Right

Attributs CSS pour déterminer où l'élément vas apparaître

Position

static : Valeur par défaut (top, left, bottom, right n'auront aucun effet)

relative : L'élément est positionné par rapport à sa position normal

absolute : L'élément est positionné par rapport à sont parent non-static le plus proche

fixed : L'élément est positionné par rapport à la fenêtre du navigateur

Lecture Supplémentaire (en anglais)

https://alistapart.com/article/css-positioning-101